Comando Básicos

Comandos são ordens que passamos ao sistema operacional para executar uma determinada tarefa.
Cada comando tem uma função específica, devemos saber a função de cada comando e escolher o mais adequado para fazer o que desejamos, por exemplo:

- `ls` -Mostra arquivos de diretórios
- `cd`- Para mudar de diretório

Esta apostila tem uma lista de vários comandos organizados por categoria com a explicação sobre o seu funcionamento e as opções aceitas (incluindo alguns exemplos).
É sempre usado um espaço depois do comando para separa-lo de uma opção ou parâmetro que será passado para o processamento. Um comando pode receber opções e parâmetros:

Opções

As opções são usadas para controlar como o comando será executado, por exemplo, para fazer uma listagem mostrando o dono, grupo, tamanho dos arquivos você deve digitar ls -l.

Opções podem ser passadas ao comando através de um "-" ou "--":

-

Opção identificada por uma letra. Podem ser usadas mais de uma opção com um único hifen. O comando `ls -l –a` é a mesma coisa de `ls –la`

--

Opção identificada por um nome. O comando `ls --al` é equivalente a `ls -a`.

Pode ser usado tanto "-" como "--", mas há casos em que somente "-" ou "--" esta disponível.

parâmetros

Um parâmetro identifica o caminho, origem, destino, entrada padrão ou saída padrão que será passada ao comando.

Se você digitar: `ls /usr/doc/copyright` , /usr/doc/copyright será o parâmetro passado ao comando ls, neste caso queremos que ele liste os arquivos do diretório /usr/doc/copyright.

É normal errar o nome de comandos, mas não se preocupe, quando isto acontecer o sistema mostrará a mensagem command not found (comando não encontrado) e voltará ao aviso de comando. As mensagens de erro não fazem nenhum mal ao seu sistema! somente dizem que algo deu errado para que você possa corrigir e entender o que aconteceu. No Linux, você tem a possibilidade de criar comandos personalizados usando outros comandos mais simples (isto será visto mais adiante). Os comandos se encaixam em duas categorias: Comandos Internos e Comandos Externos.

Por exemplo: "ls -la /usr/doc", ls é o comando, -la é a opção passada ao comando, e /usr/doc é o diretório passado como parâmetro ao comando ls.

Comandos Internos

São comandos que estão localizados dentro do interpretador de comandos (normalmente o Bash) e não no disco. Eles são carregados na memória RAM do computador junto com o interpretador de comandos.
Quando executa um comando, o interpretador de comandos verifica primeiro se ele é um Comando Interno caso não seja é verificado se é um Comando Externo.
Exemplos de comandos internos são: cd, exit, echo, bg, fg, source, help

---

Comandos Externos

São comandos que estão localizados no disco. Os comandos são procurados no disco usando o path e executados assim que encontrados.

Comandos para manipulação de diretório

- ls
- cd
- pwd
- mkdir
- rmdir

Comandos para manipulação de Arquivos

- cat
- rm
- cp
- mv

Comandos Diversos

- clear
- date
- df
- ln
- find
- free
- grep
- more
- less
- sort
- tail

- touch
- echo
- su
- sync
- uname
- shutdown
- dirname

Comandos de rede

- who
- Telnet
- finger
- ftp
- whoami
- dnsdomainname
- hostname
- talk
- ping
- rlogin
- rsh
- w
- traceroute
- netstat
- wall

Comandos para manipulação de contas

- adduser
- addgroup
- passwd
- newgrp
- userdel
- groupdel
- lastlog
- last
- Adicionando um novo grupo a um usuário
- id
- logname
- users
- groups

Configuração Geral do Sistema

Explicação do utilitário Linuxconf

- Configuração de sistema, grupos e usuarios, rotas, dns, dhcp, firewall, etc...

Explicação do utilitário Webmin

- Possue a mesma configuração acima, porém administrada via WEB.

Explicação do utilitário RPM

⇒ `rpm –qa | sort | less` → mostra todos os arquivos **rpm**'s do sistema.

⇒ `rpm –qi “nome do pacote rpm”` → mostra informações sobre o pacote **rpm** instalado.

⇒ `rpm –ql “nome do pacote rpm”` → mostra os arquivos que compõe o pacote instalado.

⇒ `rpm –qa | grep “nome do pacote rpm”` → procura pacotes que contém este nome.

⇒ `rpm –e “nome do pacote rpm”` → desinstala o pacote relacionado.

⇒ `rpm –e -–nodeps “nome do pacote rpm”` → desinstala o pacote relacionado mesmo que ele tenha dependências de outros pacotes.

⇒ `rpm –qip “nome do pacote rpm”` → obtém informações de um pacote não instalado.

⇒ `rpm –qlp “nome do pacote rpm”` → obtém informações sobre os arquivos que compõe o pacote a ser instalado.

⇒ `rpm –ivh “nome do pacote rpm”` → para instalar um pacote.

⇒ `rpm –Uvh “nome do pacote rpm”` → para atualizar um pacote.

⇒ `rpm –Va “nome do pacote”` → obtém informações sobre a situação do pacote.

Estrutura de Arquivos e Diretórios

Arquivos

É onde gravamos nossos dados. Um arquivo pode conter um texto feito por nós, uma música, programa, planilha, etc.
Cada arquivo deve ser identificado por um nome, assim ele pode ser encontrado facilmente quando desejar usa-lo. Se estiver fazendo um trabalho de história, nada melhor que salva-lo com o nome historia. Um arquivo pode ser binário ou texto.

O Linux é Case Sensitive ou seja, ele diferencia letras maiúsculas e minúsculas nos arquivos. O arquivo historia é completamente diferente de Historia. Esta regra também é válido para os comandos e diretórios. Prefira, sempre que possível, usar letras minúsculas para identificar seus arquivos, pois quase todos os comandos do sistema estão em minúsculas.
Um arquivo oculto no Linux é identificado por um "." no inicio do nome (por exemplo, .bashrc). Arquivos ocultos não aparecem em listagens normais de diretórios, deve ser usado o comando ls -a para também listar arquivos ocultos.

Extensão de arquivos

A extensão serve para identificar o tipo do arquivo. A extensão são as letras após um "." no nome de um arquivo, explicando melhor:

- relatorio**.txt** - O .txt indica que o conteúdo é um arquivo texto.
- script**.sh** - Arquivo de Script (interpretado por /bin/sh).
- system**.log** - Registro de algum programa no sistema.
- Arquivo.tar - Arquivo compactado pelo utilitário tar.
- arquivo**.gz** - Arquivo compactado pelo utilitário gzip.
- index**.html** - Página de Internet (formato Hypertexto).
- Arquivo.**c** – Scripts em C que são compilados com os compiladores CC ou GCC.
- Arquivo.**pl** – Scripts em Perl interpretados pelo programa **PERL**.
- Arquirvo.**rpm** – arquivos de programas utilizados pelo interpretador **RPM**.

A extensão de um arquivo também ajuda a saber o que precisamos fazer para abri-lo. Por exemplo, o arquivo relatorio.txt é um texto simples e podemos ver seu conteúdo através do comando, já o arquivo index.html contém uma página de Internet e precisaremos de um navegador para poder visualiza-lo (como o lynx, Mosaic ou o Netscape).
A extensão (na maioria dos casos) não é requerida pelo sistema operacional Linux, mas é conveniente o seu uso para determinarmos facilmente o tipo de arquivo e que programa precisaremos usar para abri-lo.

Arquivo texto e binário

Quanto ao tipo, um arquivo pode ser de texto ou binário:

<u>texto</u>

Seu conteúdo é compreendido pelas pessoas. Um arquivo texto pode ser uma carta, um script, um programa de computador escrito pelo programador, arquivo de configuração, etc.

<u>binário</u>

Seu conteúdo somente pode ser entendido por computadores. Contém caracteres incompreensíveis para pessoas normais. Um arquivo binário é gerado através de um arquivo de programa (formato texto) através de um processo chamado de compilação. Compilação é básicamente a conversão de um programa em linguagem humana para a linguagem de máquina.

Diretório

Diretório é o local utilizado para armazenar conjuntos arquivos para melhor organização e localização. O diretório, como o arquivo, também é "Case Sensitive" (diretório /teste é completamente diferente do diretório /Teste).
Não podem existir dois arquivos com o mesmo nome em um diretório, ou um sub-diretório com um mesmo nome de um arquivo em um mesmo diretório.

Um diretório nos sistemas Linux/UNIX são especificados por uma "/" e não uma "\" como é feito no DOS.

Diretório Raíz

Este é o diretório principal do sistema. Dentro dele estão todos os diretórios do sistema. O diretório Raíz é representado por uma "/", assim se você digitar o comando cd / você estará acessando este diretório.
Nele estão localizados outros diretórios como o /bin, /sbin, /usr, /usr/local, /mnt, /tmp, /var, /home, etc. Estes são chamados de sub-diretórios pois estão dentro do diretório "/". A estrutura de diretórios e sub-diretórios pode ser identificada da seguinte maneira:

- /
- /bin
- /sbin
- /usr
- /usr/local
- /mnt
- /tmp
- /var
- /home

A estrutura de diretórios também é chamada de Árvore de Diretórios porque é parecida com uma árvore de cabeça para baixo. Cada diretório do sistema tem seus respectivos arquivos que são armazenados conforme regras definidas pela FHS (FileSystem Hierarchy Standard - Hierarquia Padrão do Sistema de Arquivos) versão 2.0, definindo que tipo de arquivo deve ser armazenado em cada diretório.

Diretório padrão

É o diretório em que nos encontramos no momento. Também é chamado de diretório atual. Você pode digitar pwd para verificar qual é seu diretório padrão.
O diretório padrão também é identificado por um . (ponto). O comando comando ls . pode ser usado para listar os arquivos do diretório atual (é claro que isto é desnecessário porque se não digitar nenhum diretório, o comando ls listará o conteúdo do diretório atual).

Diretório home

Também chamado de diretório de usuário. Em sistemas Linux cada usuário (inclusive o root) possui seu próprio diretório onde poderá armazenar seus programas e arquivos pessoais.
Este diretório está localizado em /home/[login], neste caso se o seu login for "joao" o seu diretório home será /home/joao. O diretório home também é identificado por um ~(til), você pode digitar tanto o comando ls /home/joao como ls ~ para listar os arquivos de seu diretório home.
O diretório home do usuário root (na maioria das distribuições Linux) está localizado em /root.
Dependendo de sua configuração e do número de usuários em seu sistema, o diretório de usuário pode ter a seguinte forma: /home/[1letra_do_nome]/[login], neste caso se o seu login for "joao" o seu diretório home será /home/j/joao.

Diretório Superior

O diretório superior (Upper Directory) é identificado por .. (2 pontos).
Caso estiver no diretório /usr/local e quiser listar os arquivos do diretório /usr você pode digitar, ls .. Este recurso também pode ser usado para copiar, mover arquivos/diretórios, etc.

Diretório Anterior

O diretório anterior é identificado por -. É útil para retornar ao último diretório usado.
Se estive no diretório /usr/local e digitar cd /lib, você pode retornar facilmente para o diretório /usr/local usando cd -.

Caminho na estrutura de diretórios

São os diretórios que teremos que percorrer até checar no arquivo ou diretório que que procuramos. Se desejar ver o arquivo /usr/doc/copyright/GPL você tem duas opções:

1. Mudar o diretório padrão para /usr/doc/copyright com o comando cd /usr/doc/copyright e usar o comando cat GPL
2. Usar o comando "cat" especificando o caminho completo na estrutura de diretórios e o nome de arquivo: cat /usr/doc/copyright/GPL.

As duas soluções acima permitem que você veja o arquivo GPL. A diferença entre as duas é a seguinte:

- Na primeira, você muda o diretório padrão para /usr/doc/copyright (confira digitando pwd) e depois o comando cat GPL. Você pode ver os arquivos de /usr/doc/copyright com o comando "ls".
  /usr/doc/copyright é o caminho de diretório que devemos percorrer para chegar até o arquivo GPL.
- Na segunda, é digitado o caminho completo para o "cat" localizar o arquivo GPL: cat /usr/doc/copyright/GPL. Neste caso, você continuará no diretório padrão (confira digitando pwd). Digitando ls, os arquivos do diretório atual serão listados.

O caminho de diretórios é necessário para dizer ao sistema operacional onde encontrar um arquivo na "árvore" de diretórios.

Exemplo de diretório

Um exemplo de diretório é o seu diretório de usuário, todos seus arquivos essenciais devem ser colocadas neste diretório. Um diretório pode conter outro diretório, isto é útil quando temos muitos arquivos e queremos melhorar sua organização. Abaixo um exemplo de uma empresa que precisa controlar os arquivos de Pedidos que emite para as fábricas:
/pub/vendas - diretório principal de vendas /pub/vendas/mes01-99 - diretório contendo vendas do mês 01/1999 /pub/vendas/mes02-99 - diretório contendo vendas do mês 02/1999 /pub/vendas/mes03-99 - diretório contendo vendas do mês 03/1999

- o diretório vendas é o diretório principal.
- mes01-99 subdiretório que contém os arquivos de vendas do mês 01/1999.
- mes02-99 subdiretório que contém os arquivos de vendas do mês 02/1999.
- mes03-99 subdiretório que contém os arquivos de vendas do mês 03/1999.

mes01-99, mes02-99, mes03-99 são diretórios usados para armazenar os arquivos de pedidos do mês e ano correspondente. Isto é essencial para organização, pois se todos os pedidos fossem colocados diretamente no diretório vendas, seria muito difícil encontrar o arquivo do cliente "João" ;-)
Você deve ter reparado que usei a palavra sub-diretório para mes01-99, mes02-99 e mes03-99, porque que eles estão dentro do diretório vendas. Da mesma forma, vendas é um sub-diretório de pub.

Estrutura básica de diretórios do Sistema Linux

O sistema Linux possui a seguinte estrutura básica de diretórios:

/bin

Contém arquivos programas do sistema que são usados com frequência pelos usuários.

/boot

Contém arquivos necessários para a inicialização do sistema.

/mnt/cdrom

Ponto de montagem da unidade de CD-ROM.

/dev

Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.

/etc

Arquivos de configuração de seu computador local.

/mnt/floppy

Ponto de montagem de unidade de disquetes

/home

Diretórios contendo os arquivos dos usuários.

/lib

Bibliotecas compartilhadas pelos programas do sistema e módulos do kernel.

/lost+found

Local para a gravação de arquivos/diretórios recuperados pelo utilitário fsck.ext2. Cada partição possui seu próprio diretório lost+found.

/mnt

Ponto de montagem temporário.

/proc

Sistema de arquivos do kernel. Este diretório não existe em seu disco rígido, ele é colocado lá pelo kernel e usado por diversos programas que fazem sua leitura, verificam

configurações do sistema ou modificar o funcionamento de dispositivos do sistema através da alteração em seus arquivos.

/root

Diretório do usuário root.

/sbin

Diretório de programas usados pelo superusuário (root) para administração e controle do funcionamento do sistema.

/tmp

Diretório para armazenamento de arquivos temporários criados por programas.

/usr

Contém maior parte de seus programas. Normalmente acessível somente como leitura.

/var

Contém maior parte dos arquivos que são gravados com frequência pelos programas do sistema, e-mails, spool de impressora, cache, etc.

Sistema de Arquivos e Diretórios

No Unix/Linux, os arquivos e diretórios pode ter o tamanho de até 255 letras. Você pode identifica-lo com uma extensão (um conjunto de letras separadas do nome do arquivo por um ".").
Os programas executáveis do Linux, ao contrário dos programas de DOS e Windows, não são executados a partir de extensões .exe, .com ou .bat. O Linux (como todos os sistemas POSIX) usa a permissão de execução de arquivo para identificar se um arquivo pode ou não ser executado.
No exemplo anterior, nosso trabalho de história pode ser identificado mais facilmente caso fosse gravado com o nome trabalho.text ou trabalho.txt. Também é permitido gravar o arquivo com o nome Trabalho de Historia.txt mas não é recomendado gravar nomes de arquivos e diretórios com espaços. Porque será necessário colocar o nome do arquivo entre "aspas" para acessa-lo (por exemplo, cat "Trabalho de Historia.txt"). Ao invés de usar espaços, prefira capitalizar o arquivo (usar letras maiúsculas e minúsculas para identifica-lo): TrabalhodeHistoria.txt. Falaremos agora sobre as permissões para os arquivos e diretórios.

- Donos, grupos e outros usuários
- Tipos de Permissões de acesso
- Etapas para acesso a um arquivo/diretório
- Exemplos práticos de permissões de acesso
    - Exemplo de acesso a um arquivo
    - Exemplo de acesso a um diretório
- Permissões de Acesso Especiais
- A conta root
- chmod
- chgrp

- chown
- Modo de permissão octal

Donos, grupos e outros usuários

O principio da segurança no sistema de arquivos Linux é definir o acesso aos arquivos por donos, grupos e outros usuários:

dono

É a pessoa que criou o arquivo ou o diretório. O nome do dono do arquivo/diretório é o mesmo do usuário usado para entrar o sistema Linux. Somente o dono pode modificar as permissões de acesso do arquivo.

As permissões de acesso do dono de um arquivo somente se aplicam ao dono do arquivo/diretório. A identifição do dono também é chamada de user id (UID).

A identificação de usuário e o nome do grupo que pertence são armazenadas respectivamente nos arquivos /etc/passwd e /etc/group. Estes são arquivos textos comuns e podem ser editados em qualquer editor de texto, mas tenha cuidado para não modificar o campo que contém a senha do usuário encriptada (que pode estar armazenada neste arquivo caso não estiver usando senhas ocultas).

grupo

Para permitir que vários usuários diferentes tivessem acesso a um mesmo arquivo (já que somente o dono poderia ter acesso ao arquivo), este recurso foi criado. Cada usuário pode fazer parte de um ou mais grupos e então acessar arquivos que pertençam ao mesmo grupo que o seu (mesmo que estes arquivos tenham outro dono).

Por padrão, quando um novo usuário é criado, o grupo ele pertencerá será o mesmo de seu grupo primário (exceto pelas condições que explicarei adiante) (veja isto através do comando id. A identificação do grupo é chamada de gid (group id).

outros

É a categoria de usuários que não são donos ou não pertencem ao grupo do arquivo.

Tipos de Permissões de acesso

Quanto aos tipos de permissões que se aplicam ao dono, grupo e outros usuários, temos 3 permissões básicas:

- r - Permissão de leitura para arquivos. Caso for um diretório, permite listar seu conteúdo (através do comando ls, por exemplo).
- w - Permissão de gravação para arquivos. Caso for um diretório, permite a gravação de arquivos ou outros diretórios dentro dele.
  Para que um arquivo/diretório possa ser apagado, é necessário o acesso a gravação.
- x - Permite executar um arquivo (caso seja um programa executável). Caso seja um diretório, permite que seja acessado através do comando cd.

As permissões de acesso a um arquivo/diretório podem ser visualizadas com o uso do comando ls -la. As 3 letras (rwx) são agrupadas da seguinte forma:

```
-rwxrwxrwx  gleydson  users  teste
```

Virou uma bagunção não? Vou explicar cada parte para entender o que quer dizer as 10 letras acima (da esquerda para a direita):

- A primeira letra diz qual é o tipo do arquivo. Caso tiver um "d" é um diretório, um "l" um link a um arquivo no sistema, um "-" quer dizer que é um arquivo comum, etc.
- Da segunda a quarta letra (rwx) dizem qual é a permissão de acesso ao dono do arquivo. Neste caso gleydson ele tem a permissão de ler(r - read), gravar (w - write) e executar (x - execute) o arquivo teste.
- Da quinta a sétima letra (rwx) diz qual é a permissão de acesso ao grupo do arquivo. Neste caso todos os usuários que pertencem ao grupo users tem a permissão de ler (r), gravar (w), e também executar (x) o arquivo teste
- Da oitava a décima letra (rwx) diz qual é a permissão de acesso para os outros usuários. Neste caso todos os usuários que não são donos do arquivo teste tem a permissão para ler, gravar e executar o programa.

Veja o comando chmod para detalhes sobre a mudança das permissões de acesso de arquivos/diretórios.

Etapas para acesso a um arquivo/diretório

O acesso a um arquivo/diretório é feito verificando primeiro se o usuário que acessará o arquivo é o seu dono, caso seja, as permissões de dono do arquivo são aplicadas. Caso não seja o dono do arquivo/diretório, é verificado se ele pertence ao grupo correspondente, caso pertença, as permissões do grupo são aplicadas. Caso não pertença ao grupo, são verificadas as permissões de acesso para os outros usuários que não são donos e não pertencem ao grupo correspondente ao arquivo/diretório.

Após verificar aonde o usuário se encaixa nas permissões de acesso do arquivo (se ele é o dono, pertence ao grupo, ou outros usuários), é verificado se ele terá permissão acesso para o que deseja fazer (ler, gravar ou executar o arquivo), caso não tenha, o acesso é negado, mostrando uma mensagem do tipo: "Permission denied" (permissão negada).

O que isto que dizer é que mesmo que você seja o dono do arquivo e definir o acesso do dono (através do comando chmod) como somente leitura (r) mas o acesso dos outros usuários como leitura e gravação, você somente poderá ler este arquivo mas os outros usuários poderão ler/grava-lo.

As permissões de acesso (leitura, gravação, execução) para donos, grupos e outros usuários são independentes, permitindo assim um nível de acesso diferenciado. Lembre-se: Somente o dono pode modificar um arquivo/diretório!

Exemplos práticos de permissões de acesso

Abaixo dois exemplos práticos de permissão de acesso: Exemplo de acesso a um arquivo e Exemplo de acesso a um diretório. Os dois exemplos são explicados passo a passo para uma perfeita compreensão do assunto.

Abaixo um exemplo e explicação das permissões de acesso a um arquivo no `Linux` (obtido com o comando `ls -la`, explicarei passo a passo cada parte:

-rwxr-xr-- 1 gleydson user 8192 nov 4 16:00 teste

-rwxr-xr--

Estas são as permissões de acesso ao arquivo teste. Um conjunto de 10 letras que especificam o tipo do arquivo, permissão do dono do arquivo, grupo do arquivo e outros usuários. Veja a explicação detalhada sobre cada uma abaixo:

- rwxr-xr--

A primeira letra (do conjunto das 10 letras) determina o tipo do arquivos. Se a letra for um **d** é um diretório, e você poderá acessa-lo usando o comando cd. Caso for um **l** é um link simbólico para algum arquivo ou diretório no sistema . Um significa que é um arquivo normal.

-**rwx**r-xr--

Estas 3 letras (da segunda a quarta do conjunto das 10 letras) são as permissões de acesso do dono do arquivo teste. O dono (neste caso gleydson) tem a permissão para ler(r), gravar(w) e executar (x) o arquivo teste.

-rwx**r-x**r--

Estas 3 letras (da quinta a sétima do conjunto das 10 letras) são as permissões de acesso dos usuários que pertencem ao grupo user do arquivo teste. Os usuários que pertencem ao grupo user tem a permissão somente para ler(r) e executar(x) o arquivo teste não podendo modifica-lo ou apaga-lo.

-rwxr-x**r--**

Estas 3 letras (da oitava a décima) são as permissões de acesso para usuários que **não** são donos do arquivo teste e que **não** pertencem ao grupo user. Neste caso, estas pessoas somente terão a permissão para ver o conteúdo do arquivo teste.

gleydson

Nome do dono do arquivo teste.

user

Nome do grupo que o arquivo teste pertence.

teste

Nome do arquivo.

Exemplo de acesso a um diretório

Abaixo um exemplo com explicações das permissões de acesso a um diretório no Linux:

drwxr-x--- 2 gleydson user 1024 nov 4 17:55 exemplo

drwxr-x---

Permissões de acesso ao diretório exemplo. É um conjunto de 10 letras que especificam o tipo de arquivo, permissão do dono do diretório, grupo que o diretório pertence e permissão de acesso a outros usuários. Veja as explicações abaixo:

drwxr-x---

A primeira letra (do conjunto das 10) determina o tipo do arquivo. Neste caso é um diretório porque tem a letra **d**.

d**rwx**r-x---

Estas 3 letras (da segunda a quarta) são as permissões de acesso do dono do diretório exemplo. O dono do diretório (neste caso gleydson) tem a permissão para listar arquivos do diretório(r), gravar arquivos no diretório(w) e entrar no diretório(x).

drwx**r-x**---

Estas 3 letras (da quinta a sétima) são as permissões de acesso dos usuários que pertencem ao grupo user. Os usuários que pertencem ao grupo user tem a permissão somente para listar arquivos do diretório(r) e entrar no diretório(x) exemplo.

drwxr-x**---**

Estas 3 letras (da oitava a décima) são as permissões de acesso para usuários que **não** são donos do diretório exemplo e que **não** pertencem ao grupo user. Com as permissões acima, nenhum usuário que se encaixe nas condições de dono e grupo do diretório tem a permissão de acessa-lo.

gleydson

Nome do dono do diretório exemplo

user

Nome do grupo que diretório exemplo pertence.

exemplo

Nome do diretório.

Exemplo de acesso a um arquivo

Abaixo um exemplo e explicação das permissões de acesso a um arquivo no Linux (obtido com o comando ls -la, explicarei passo a passo cada parte:
-rwxr-xr-- 1 gleydson user 8192 nov 4 16:00 teste

-rwxr-xr--

Estas são as permissões de acesso ao arquivo teste. Um conjunto de 10 letras que especificam o tipo do arquivo, permissão do dono do arquivo, grupo do arquivo e outros usuários. Veja a explicação detalhada sobre cada uma abaixo:

- rwxr-xr--

A primeira letra (do conjunto das 10 letras) determina o tipo do arquivos. Se a letra for um **d** é um diretório, e você poderá acessa-lo usando o comando cd. Caso for um **l** é um link simbólico para algum arquivo ou diretório no sistema . Um **-** significa que é um arquivo normal.

-**rwx**r-xr--

Estas 3 letras (da segunda a quarta do conjunto das 10 letras) são as permissões de acesso do dono do arquivo teste. O dono (neste caso gleydson) tem a permissão para ler(r), gravar(w) e executar (x) o arquivo teste.

-rwx**r-x**r--

Estas 3 letras (da quinta a sétima do conjunto das 10 letras) são as permissões de acesso dos usuários que pertencem ao grupo user do arquivo teste. Os usuários que pertencem ao grupo user tem a permissão somente para ler(r) e executar(x) o arquivo teste não podendo modifica-lo ou apaga-lo.

-rwxr-x**r--**

Estas 3 letras (da oitava a décima) são as permissões de acesso para usuários que **não** são donos do arquivo teste e que **não** pertencem ao grupo user. Neste caso, estas pessoas somente terão a permissão para ver o conteúdo do arquivo teste.

gleydson

Nome do dono do arquivo teste.

user

Nome do grupo que o arquivo teste pertence.

teste

Nome do arquivo.

Exemplo de acesso a um diretório

Abaixo um exemplo com explicações das permissões de acesso a um diretório no Linux:
drwxr-x--- 2 gleydson user 1024 nov 4 17:55 exemplo

drwxr-x---

Permissões de acesso ao diretório exemplo. É um conjunto de 10 letras que especificam o tipo de arquivo, permissão do dono do diretório, grupo que o diretório pertence e permissão de acesso a outros usuários. Veja as explicações abaixo:

drwxr-x---

A primeira letra (do conjunto das 10) determina o tipo do arquivo. Neste caso é um diretório porque tem a letra **d**.

d**rwx**r-x---

Estas 3 letras (da segunda a quarta) são as permissões de acesso do dono do diretório exemplo. O dono do diretório (neste caso gleydson) tem a permissão para listar arquivos do diretório(r), gravar arquivos no diretório(w) e entrar no diretório(x).

drwx**r-x**---

Estas 3 letras (da quinta a sétima) são as permissões de acesso dos usuários que pertencem ao grupo user. Os usuários que pertencem ao grupo user tem a permissão somente para listar arquivos do diretório(r) e entrar no diretório(x) exemplo.

drwxr-x**---**

Estas 3 letras (da oitava a décima) são as permissões de acesso para usuários que **não** são donos do diretório exemplo e que **não** pertencem ao grupo user. Com as permissões acima, nenhum usuário que se encaixe nas condições de dono e grupo do diretório tem a permissão de acessa-lo.

gleydson

Nome do dono do diretório exemplo

user

Nome do grupo que diretório exemplo pertence.

exemplo

Nome do diretório.

OBSERVAÇÕES:

- O usuário root não tem nenhuma restrição de acesso ao sistema.
- Se você tem permissões de gravação no diretório e tentar apagar um arquivo que você não tem permissão de gravação, o sistema perguntará se você confirma a exclusão do

arquivo apesar do modo leitura. Caso você tenha permissões de gravação no arquivo, o arquivo será apagado por padrão sem mostrar nenhuma mensagem de erro (a não ser que seja especificada a opção -i com o comando rm.)

- Por outro lado, mesmo que você tenha permissões de gravação em um arquivo mas não tenha permissões de gravação em um diretório, a exclusão do arquivo será negada!.

Isto mostra que é levado mais em consideração a permissão de acesso do diretório do que as permissões dos arquivos e sub-diretórios que ele contém. Este ponto é muitas vezes ignorado por muitas pessoas e expõem seu sistema a riscos de segurança. Imagine o problema que algum usuário que não tenha permissão de gravação em um arquivo mas que a tenha no diretório pode causar em um sistema mal administrado.

Permissões de Acesso Especiais

Em adição as três permissões básicas (rwx), existem permissões de acesso especiais (stX) que afetam arquivos executáveis e diretórios:

- s - Quando é usado na permissão de acesso do Dono, ajusta a identificação efetiva usuário do processo durante a execução de um programa, também chamado de bit setuid. Não tem efeito em diretórios.
  Quando s é usado na permissão de acesso do Grupo, ajusta a identificação efetiva do grupo do processo durante a execução de um programa, chamado de bit setgid. É identificado pela letra s no lugar da permissão de execução do grupo do arquivo/diretório. Em diretórios, força que os arquivos criados dentro dele pertençam ao mesmo grupo do diretório, ao invés do grupo primário que o usuário pertence.
  Ambos setgid e setuid podem aparecer ao mesmo tempo no mesmo arquivo/diretório. A permissão de acesso especial s somente pode aparecer no campo Dono e Grupo.
- t - Salva a imagem do texto do programa no dispositivo swap, assim ele será carregado mais rápidamente quando executado, também chamado de stick bit.
  Em diretórios, impede que outros usuários removam arquivos dos quais não são donos. Isto é chamado de colocar o diretório em modo append-only. Um exemplo de diretório que se encaixa perfeitamente nesta condição é o /tmp, todos os usuários devem ter acesso para que seus programas possam criar os arquivos temporários lá, mas nenhum pode apagar arquivos dos outros. A permissão especial t, pode ser especificada somente no campo outros usuários das permissões de acesso.
- X - Se você usar X ao invés de x, a permissão de execução somente é afetada se o arquivo ja tiver permissões de execução. Em diretórios ela tem o mesmo efeito que a permissão de execução x.
- Exemplo da permissão de acesso especial X:
  3. Crie um arquivo teste (digitando touch teste) e defina sua permissão para rw-rw-r-- (chmod ug=rw,o=r teste ou chmod 664 teste).
  4. Agora use o comando chmod a+X teste
  5. digite ls -l
  6. Veja que as permissões do arquivo não foram afetadas
  7. agora digite chmod o+x teste
  8. digite ls -l, você colocou a permissão de execução para os outros usuários
  9. Agora use novamente o comando chmod a+X teste
  10. digite ls -l

11. Veja que agora a permissão de execução foi concedida a todos os usuários, pois foi verificado que o arquivo era executável (tinha permissão de execução para outros usuários).
12. Agora use o comando chmod a-X teste
13. Ele também funcionará e removerá as permissões de execução de todos os usuários, porque o arquivo teste tem permissão de execução (confira digitando ls -l).
14. Agora tente novamente o chmod a+X teste
15. Você deve ter reparado que a permissão de acesso especial X é semelhante a x, mas somente faz efeito quanto o arquivo já tem permissão de execução para o dono, grupo ou outros usuários.

Em diretórios, a permissão de acesso especial X funciona da mesma forma que x, até mesmo se o diretório não tiver nenhuma permissão de acesso (x).

A conta root

Esta seção foi retirada do Manual de Instalação da Debian.

A conta root é também chamada de super usuário, este é um login que não possui restrições de segurança. A conta root somente deve ser usada para fazer a administração do sistema, e usada o menor tempo possível.

Qualquer senha que criar deverá conter de 6 a 8 caracteres, e também poderá conter letras maiúsculas e minúsculas, e também caracteres de pontuação. Tenha um cuidado especial quando escolher sua senha root, porque ela é a conta mais poderosa. Evite palavras de dicionário ou o uso de qualquer outros dados pessoais que podem ser adivinhados.

Se qualquer um lhe pedir senha root, seja extremamente cuidadoso. Você normalmente nunca deve distribuir sua conta root, a não ser que esteja administrando um computador com mais de um administrador do sistema.

Utilize uma conta de usuário normal ao invés da conta root para operar seu sistema. Porque não usar a conta root? Bem, uma razão para evitar usar privilégios root é por causa da facilidade de se cometer danos irreparáveis como root. Outra razão é que você pode ser enganado e rodar um programa Cavalo de Tróia -- que é um programa que obtém poderes do super usuário para comprometer a segurança do seu sistema sem que você saiba.

---

chmod

Muda a permissão de acesso a um arquivo ou diretório. Com este comando você pode escolher se usuário ou grupo terá permissões para ler, gravar, executar um arquivo ou arquivos. Sempre que um arquivo é criado, seu dono é o usuário que o criou e seu grupo é o grupo do usuário (exceto para diretórios configurados com a permissão de grupo "s", será visto adiante).

```
chmod [opções] [permissões] [diretório/arquivo]
```

Onde:

diretório/arquivo

Diretório ou arquivo que terá sua permissão mudada

opções

-v, --verbose

Mostra todos os arquivos que estão sendo processados.

-f, --silent

Não mostra a maior parte das mensagens de erro

-c, --change

Semelhante a opção -v, mas só mostra os arquivos que tiveram ae permissões mudadas.

-R, --recursive

Muda permissões de acesso do diretório/arquivo no diretório atual e sub-diretórios.

ugoa+-=rwxXst

- ugoa - Controla que nível de acesso será mudado. Especificam, em ordem, usuário(u), grupo(g), outros(o), todos(a).
- +-= - + coloca a permissão, - retira a permissão do arquivo e = define a permissão exatamente como especificado.
- rwx - r permissão de leitura do arquivo. w permissão de gravação. x permissão de execução (ou acesso a diretórios).

chmod não muda permissões de links simbólicos, as permissões devem ser mudadas no arquivo alvo do link. Também podem ser usados códigos numéricos octais para a mudança das permissões de acesso a arquivos/diretórios.

DICA: É possível copiar permissões de acesso do arquivo/diretório, por exemplo, se o arquivo teste.txt tiver a permissão de acesso r-xr----- e você digitar chmod o=u, as permissões de acesso dos outros usuários (o) serão idênticas ao do dono (u). Então a nova permissão de acesso do arquivo teste.txt será r-xr--r-x

Exemplos de permissões de acesso:

```
chmod g+r *
```

Permite que todos os usuários que pertençam ao grupo dos arquivos(g) tenham(+) permissões de leitura(r) em todos os arquivos do diretório atual.

```
chmod o-r teste.txt
```

Retira(-) a permissão de leitura(r) do arquivo teste.txt para os outros usuários (usuários que não são donos e não pertencem ao grupo do arquivo teste.txt).

```
chmod uo+x teste.txt
```

Inclui (+) a permissão de execução do arquivo teste.txt para o dono e grupo do arquivo.

```
chmod a+x teste.txt
```

Inclui (+) a permissão de execução do arquivo teste.txt para o dono, grupo e outros usuários.

```
chmod a=rw teste.txt
```

Define a permissão de todos os usuários exatamente (=) para leitura e gravação do arquivo teste.txt.

---

chgrp

Muda o grupo de um arquivo/diretório.

```
chgrp [opções] [grupo] [arquivo/diretório]
```

Onde:

grupo

Novo grupo do arquivo/diretório

arquivo/diretório

Arquivo/diretório que terá o grupo alterado.

opções

-c, --changes

Somente mostra os arquivos/grupos que forem alterados.

-f, --silent

Não mostra mensagens de erro para arquivos/diretórios que não puderam ser alterados.

-v, --verbose

Mostra todas as mensagens e arquivos sendo modificados.

-R, --recursive

Altera os grupos de arquivos/sub-diretórios do diretório atual.

---

chown

Muda dono de um arquivo/diretório. Opcionalmente pode também ser usado para mudar o grupo.

```
chown [opções] [dono.grupo] [diretório/arquivo]
```

onde:

dono.grupo

Nome do dono.grupo que será atribuído ao diretório/arquivo. O grupo é opcional.

diretório/arquivo

Diretório/arquivo que o dono.grupo será modificado.

opções

-v, --verbose

Mostra os arquivos enquanto são alterados.

-f, --supress

Não mostra mensagens de erro durante a execução do programa.

-c, --changes

Mostra somente arquivos que forem alterados.

-R, --recursive

Altera dono e grupo de arquivos no diretório atual e sub-diretórios.

O dono.grupo pode ser especificado usando o nome de grupo ou o código numérico correspondente ao grupo (GID).
Você deve ter permissões de gravação no diretório/arquivo para alterar seu dono/grupo.

- `chown joao teste.txt` - Muda o dono do arquivo teste.txt para joao.
- `chown joao.users teste.txt` - Muda o dono do arquivo teste.txt para joao e seu grupo para users.
- `chown -R joao.users *` - Muda o dono/grupo dos arquivos do diretório atual e sub-diretórios para joao/users (desde que você tenha permissões de gravação no diretórios e sub-diretórios).

Modo de permissão octal

Ao invés de utilizar os modos de permissão +r, -r, etc, pode ser usado o modo octal para se alterar a permissão de acesso a um arquivo. O modo octal é um conjunto de oito números onde cada número define um tipo de acesso diferente.
É mais flexível gerenciar permissões de acesso usando o modo octal ao invés do comum, pois você especifica diretamente a permissão do dono, grupo, outros ao invés de gerenciar as permissões de cada um separadamente. Abaixo a lista de permissões de acesso octal:

- 0 - Nenhuma permissão de acesso. Equivalente a -rwx
- 1 - Permissão de execução (x).
- 2 - Permissão de gravação (w).
- 3 - Permissão de gravação e execução (wx).
- 4 - Permissão de leitura (r).

- 5 - Permissão de leitura e execução (rx).
- 6 - Permissão de leitura e gravação (rw).
- 7 - Permissão de leitura, gravação e execução. Equivalente a +rwx

O uso de um deste números define a permissão de acesso do dono, grupo ou outros usuários. Um modo fácil de entender como as permissões de acesso octais funcionam, é através da seguinte tabela:

- 1 = Executar
- 2 = Gravar
- 4 = Ler

* Para Dono e Grupo, multiplique as permissões acima por x100 e x10
e para as permissões de acesso especiais:
1000 = Salva imagem do texto no dispositivo de troca
2000 = Ajusta o bit setgid na execução
4000 = Ajusta o bit setuid na execução
Basta agora fazer o seguinte:

- Somente permissão de execução, use 1
- Somente a permissão de leitura, use 4
- Somente permissão de gravação, use 2
- Permissão de leitura/gravação, use 6 (equivale a 2+4 / Gravar+Ler)
- Permissão de leitura/execução, use 5 (equivale a 1+4 / Executar+Ler)
- Permissão de execução/gravação, use 3 (equivale a 1+2 / Executar+Gravar)
- Permissão de leitura/gravação/execução, use 7 (equivale a 1+2+4 / Executar+Gravar+Ler)
- Salvar texto no dispositivo de troca, use 1000
- Ajustar bit setgid, use 2000
- Ajustar bip setuid, use 4000
- Salvar texto e ajustar bit setuid, use 5000 (equivale a 1000+4000 / Salvar texto + bit setuid)
- Ajustar bit setuid e setgid, use 6000 (equivale a 4000+2000 / setuid + setgid)

Vamos a prática com alguns exemplos:

"chmod 764 teste"

Os números são interpretados da **direita para a esquerda** como permissão de acesso aos outros usuários (4), grupo (6), e dono (7). O exemplo acima faz os outros usuários (4) terem acesso somente leitura (r) ao arquivo teste, o grupo (6) ter a permissão de leitura e gravação (w), e o dono (7) ter permissão de leitura, gravação e execução (rwx) ao arquivo teste.
Outro exemplo:

"chmod 40 teste"

O exemplo acima define a permissão de acesso dos outros usuários (0) como nenhuma, e define a permissão de acesso do grupo (4) como somente leitura (r). Note usei somente dois números

e então a permissão de acesso do dono do arquivo não é modificada (leia as permissões de acesso da direita para a esquerda!).

"chmod 752 teste"

O exemplo acima define a permissão de acesso dos outros usuários (2) para somente execução (x), o acesso do grupo(5) como leitura e execução (rx) e o acesso do dono(7) como leitura, gravação e execução (rwx).

"chmod 4752 teste"

O exemplo acima define a permissão de acesso dos outros usuários (2) para somente execução (x), acesso do grupo (4) como leitura e execução (rx), o acesso do dono (7) como leitura, gravação e execução (rwx) e ajusta o bit setgid (4) para o arquivo teste

Adminstração do Sistema

Utilizando o comando **top** para acompanhar o andamento do processos e gerenciamento de memória do sistema.

Utilizando o comando **ps** para visualizar os processos atuais do sistema

Utilizando o comando kill, kill –9, killall e kill –HUP para finalizar processos ou reinicializar processos.

Como fazer backup e restore do sistema UNIX/LINUX usando os utilitários **cpio** e **tar**.

Análize de log's

Utilizando o **logcheck** que envia um E-Mail periodicamente ao usuário alertando sobre ocorrências especiais encontradas nos logs do sistema, como tentativas de invasão sem sucesso, tentativas de acesso ao usuário root do sistema, erros nos dispositivos, mensagens dos daemons, inetd, etc

Manutenção do Sistema

- Checagem dos sistemas de arquivos
    - fsck.ext2
- fsck.minix
- badblocks
- defrag
- Limpando arquivos de LOGS
- Tarefas automáticas de manutenção do sistema
- cron
- O formato de um arquivo crontab

Modo Texto

Interpretador de comandos

Também conhecido como "shell". É o programa responsável em interpretar as instruções enviadas pelo usuário e seus programas ao sistema operacional (o kernel). Ele que executa comandos lidos do dispositivo de entrada padrão (teclado) ou de um arquivo executável. É a principal ligação entre o usuário, os programas e o kernel. O Linux possui diversos tipos de interpretadores de comandos, entre eles posso destacar o bash, ash, csh, tcsh, sh, etc. Entre eles o mais usado é o bash. O interpretador de comandos do DOS, por exemplo, é o command.com.
Os comandos podem ser enviados de duas maneiras para o interpretador: interativa e não-interativa:

Interativa

Os comandos são digitados no aviso de comando e passados ao interpretador de comandos um a um. Neste modo, o computador depende do usuário para executar uma tarefa, ou próximo comando.

Não-interativa

São usados arquivos de comandos criados pelo usuário (scripts) para o computador executar os comandos na ordem encontrada no arquivo. Neste modo, o computador executa os comandos do arquivo um por um e dependendo do término do comando, o script pode checar qual será o próximo comando que será executado e dar continuidade ao processamento.
Este sistema é útil quando temos que digitar por várias vezes seguidas um mesmo comando ou para compilar algum programa complexo.
O shell Bash possui ainda outra característica interessante: A completação dos nomes de comandos. Isto é feito pressionando-se a tecla TAB, o comando é completado e acrescentado um espaço. Isto funciona sem problemas para comandos internos, caso o comando não seja encontrado, o Bash emite um beep.
Exemplo: ech (pressione TAB).

Editores de Texto

- vi

Modo Texto - (existem algumas versões adaptadas para o modo gráfico). É um dos editores padrões dos sistemas Linux e sua interface é complexa e possui muitas funções (usuários Linux avançados adoram a quantidade de funções deste programa). Recomendo que aprenda o básico sobre ele, pois sempre estará disponível caso ocorra algum problema no sistema.
Para sair do editor vi sem salvar pressione ESC e digite :q!. Para sair do editor e salvar pressione ESC e digite :wq.

- elvis

Modo Texto - possui boa interface de comunicação com o usuário, suporte a HTML e Metacaracteres.

- ae

Modo Texto - é um dos editores padrões dos sistemas Linux (encontrado nas distribuições Debian e baseadas). Sua interface é mais fácil que o vi. Também recomendo que aprenda o básico sobre ele, pois é requerido para a manutenção do sistema.
Para sair do ae sem salvar pressione CTRL+Q, para salvar o texto pressione CTRL+X e CTRL+W (após isto se quiser sair do editor, pressione CTRL+Q).

- jed

Modo Texto - Recomendável para aqueles que estão acostumados com o EDIT do DOS e gostam de menus suspensos. Sua interface é de fácil operação.
O jed possui recursos poderosos para programadores de C e outras linguagens que faz auto-tabulação, auto-identação e delimitação de blocos de código através de cores.

- mcedit

Modo Texto - Muito fácil de utilizar e possui interface em Português do Brasil, em geral não requer um tutorial para aprendizado. Este programa faz parte do pacote Midnight Commander (conhecido também como mc).
Você utiliza as teclas de função (F1 a F10) para salvar o texto, procurar palavras no texto, pedir ajuda, sair, etc. Ele possui recursos para colorir blocos de código (testado com arquivos HTML e SGML).

- joe

Modo Texto - É um editor muito versátil e você pode escolher inclusive sua interface.

- gedit

Modo Gráfico - editor do Gnome, sua interface de comunicação é ótima e recomendado para aqueles que gostam de trabalhar com muitos arquivos abertos, copiar e colar, etc. Possui muitos recursos de operação de arquivo, tabulações, browser, diff de documentos, etc.

- gxedit

Modo Gráfico - Editor no estilo do gedit, sua interface de comunicação com o usuário é ótima, possui suporte a e-mail, mede o número de toques por minuto do usuário (digitação), suporte a tags HTML, audio, rede, correção ortográfica, etc.

Redirecionamento de Entrada e Saída

Esta seção explica o funcionamento dos recursos de direcionamento de entrada e saída do sistema Linux.

- ## Redirecionamento: >

Redireciona a saída de um programa/comando/script para algum dispositivo ou arquivo ao invés do dispositivo de saida padrão (tela). Quando é usado com arquivos, este redirecionamento cria ou substitui o conteúdo do arquivo. .
Por exemplo, você pode usar o comando ls para listar arquivos e usar ls >listagem para enviar a saída do comando para o arquivo listagem. Use o comando cat para visualizar o conteúdo do arquivo listagem.
O mesmo comando pode ser redirecionado para o segundo console /dev/tty2 usando: `ls >/dev/tty2`, o resultado do comando ls será mostrado no segundo console (pressione ALT e F2 para mudar para o segundo console e ALT e F1 para retornar ao primeiro).

- ## Redirecionamento: >>

Redireciona a saída de um programa/comando/script para algum dispositivo ou final de arquivo ao invés do dispositivo de saída padrão (tela). A diferença entre este redirecionamento duplo e o simples, é se caso for usado com arquivos, adiciona a saída do comando ao final do arquivo existente ao invés de substituir seu conteúdo. .
Por exemplo, você pode acrescentar a saída do comando ls ao arquivo listagem do capítulo anterior usando `ls / >>listagem`. Use o comando cat para visualizar o conteúdo do arquivo listagem.
O mesmo comando pode ser redirecionado para o segundo console /dev/tty2 usando: `ls >/dev/tty2`, o resultado do comando ls será mostrado no segundo console (pressione ALT e F2 para mudar para o segundo console e ALT e F1 para retornar ao primeiro).

- ## Redirecionamento: <

Direciona a entrada padrão de arquivo/dispositivo para um comando. Este comando faz o contrário do anterior, ele envia dados ao comando.
Você pode usar o comando `cat < teste.txt` para enviar o conteúdo do arquivo teste.txt ao comando cat que mostrará seu conteúdo (é claro que o mesmo resultado pode ser obtido com cat teste.txt mas este exemplo serviu para mostrar a funcionalidade do <).

- ## Redirecionamento: | (pipe)

Envia a saída de um comando para a entrada do próximo comando para continuidade do processamento. Os dados enviados são processados pelo próximo comando que mostrará o resultado do processamento.

Por exemplo: `ls -la|more`,, este comando faz a listagem longa de arquivos que é enviado ao comando more (que tem a função de efetuar uma pausa a cada 25 linhas do arquivo).
Outro exemplo é o comando `"locate find|grep bin/"` , neste comando todos os caminhos/arquivos que contém find na listagem serão mostrados (inclusive man pages, bibliotecas, etc.), então enviamos a saída deste comando para grep bin/ para mostrar somente os diretórios que contém binários. Mesmo assim a listagem ocupe mais de uma tela, podemos acrescentar o more: locate find|grep bin/|more.
Podem ser usados mais de um comando de redirecionamento (<, >, |) em um mesmo comando.

- ### Diferença entre o "|" e o ">"

A principal diferença entre o "|" e o ">", é que o Pipe envolve processamento entre comandos, ou seja, a saída de um comando é enviado a entrada do próximo e o ">" redireciona a saída de um comando para um arquivo/dispositivo.
Você pode notar pelo exemplo acima (ls -la|more) que ambos ls e more são comandos porque estão separados por um "|"! Se um deles não existir ou estiver digitado incorretamente, será mostrada uma mensagem de erro.
Um resultado diferente seria obtido usando um ">" no lugar do "|"; A saída do comando ls -la seria gravada em um arquivo chamado more.

- ### Redirecionamento: tee

Envia o resultado do programa para a saída padrão (tela) e para um arquivo ao mesmo tempo. Este comando deve ser usado com o pipe "|".

`comando|tee [arquivo]`

Exemplo: `ls -la|tee listagem.txt`, a saída do comando será mostrada normalmente na tela e ao mesmo tempo gravada no arquivo listagem.txt.

Linux/Dos

Este capítulo explica diferença e particularidades do sistema Linux comparado ao DOS/Windows e uma lista de equivalência entre comandos e programas DOS e Linux, que pode servir de comparação para que o usuário possa conhecer e utilizar os comandos/programas Linux que tem a mesma função no ambiente DOS/Windows.

- Quando entrar pela primeira vez no Linux (ou qualquer outro UNIX, a primeira coisa que verá será a palavra login: escrita na tela.
  A sua aventura começa aqui, você deve ser uma pessoa cadastrada no sistema (ter uma conta) para que poder entrar. No login você digita seu nome (por exemplo, gleydson) e pressiona Enter. Agora será lhe pedida a senha, repare que a senha não é mostrada enquanto é digitada, isto serve de segurança e poder enganar pessoas que estão próximas de você "tocando" algumas teclas a mais enquanto digita a senha e fazendo-as

pensar que você usa uma grande senha ;-) (com os asteriscos aparecendo isto não seria possível).
Caso cometa erros durante a digitação da senha, basta pressionar a tecla BackSpace para apagar o último caracter digitado e terminar a entrada da senha.
Pressione Enter, se tudo ocorrer bem você estará dentro do sistema e será presenteado com o símbolo # (caso tenha entrado como usuário root) ou $ (caso tenha entrado como um usuário normal).
Existe um mecanismo de segurança que te alerta sobre eventuais tentativas de entrada no sistema por intrusos usando seu login, faça um teste: entre com seu login e digite a senha errada, na segunda vez entre com a senha correta no sistema. Na penúltima linha das mensagens aparece uma mensagem "1 failure since last login", o que quer dizer "1 falha desde o último login". Isto significa que alguém tentou entrar 1 vez com seu nome e senha no sistema, sem sucesso.

- A conta root não tem restrições de acesso ao sistema e pode fazer tudo o que quiser, é equivalente ao usuário normal do DOS e Windows. Use a conta root somente para manutenções no sistema e instalação de programas, qualquer movimento errado pode comprometer todo o sistema.
- No Linux os diretório são identificados por uma / e não por uma \ como acontece no DOS. Para entrar no diretório /bin, você deve usar cd /bin.
- Os comandos são case-sensitive, o que significa que ele diferencia as letras maiúsculas de minúsculas em arquivos e diretórios. O comando ls e LS são completamente diferentes.
- A multitarefa lhe permite usar vários programas simultaneamente (não pense que multitarefa somente funciona em ambientes gráficos, pois isto é errado!). O
- Os dispositivos também são identificados e uma forma diferente que no DOS por exemplo:

DOS/Windows	Linux
A:	/dev/fd0
B:	/dev/fd1
C:	/dev/hda1
LPT1	/dev/lp0
LPT2	/dev/lp1
LPT3	/dev/lp2
COM1	/dev/ttyS0
COM2	/dev/ttyS1
COM3	/dev/ttyS2
COM4	/dev/ttyS3

- Os recursos multiusuário lhe permite acessar o sistema de qualquer lugar sem instalar nenhum driver, ou programa gigante, apenas através de conexões TCP/IP, como a Internet. Também é possível acessar o sistema localmente com vários usuários (cada um executando tarefas completamente independente dos outros) através dos Terminais Virtuais. Faça um teste: pressione ao mesmo tempo a tecla ALT e F2 e você será levado para o segundo Terminal Virtual, pressione novamente ALT e F1 para retornar ao anterior.

- Para reiniciar o computador, você pode pressionar CTRL+ALT+DEL (como usuário root) ou digitar shudown -r now. .
- Para desligar o computador, digite shutdown -h now e espere o aparecimento da mensagem Power Down para apertar o botão LIGA/DESLIGA do computador.

Comandos equivalentes entre DOS e Linux

Esta seção contém os comandos equivalentes entre estes dois sistemas e a avaliação entre ambos. Grande parte dos comandos podem ser usados da mesma forma que no DOS, mas os comandos Linux possuem avanços para utilização neste ambiente multiusuário/multitarefa.
O objetivo desta seção é permitir as pessoas com experiência em DOS fazer rapidamente no Linux as tarefas que fazem no DOS. A primeira coluna tem o nome do comando no DOS, a segunda o comando que possui a mesma função no Linux e na terceira coluna as diferenças.

```
DOS       Linux                            Diferenças
--------  ------------ ---------------------------------------------------
cls       clear        Sem diferenças
dir       ls -la       A listagem no Linux possui mais campos (as
                       permissões de acesso) e o total de espaço ocupado
                       no diretório e livre no disco deve ser visto
                       separadamente usando o comando du e df.
                       Permite também listar o conteúdo de diversos
                       diretórios com um só comando (ls /bin /sbin /...)
dir/s     ls -lR       Sem diferenças.
dir/od    ls -tr       Sem diferenças.
cd        cd           Poucas diferenças. cd sem parâmetros retorna ao
                       diretório de usuário e também permite o uso
                       de "cd -" para retornar ao diretório anteriormente
                       acessado.
del       rm           Poucas diferenças. O rm do Linux permite
                       especificar diversos arquivos que serão apagados
                       (rm arquivo1 arquivo2 arquivo3). Para ser mostrados
                       os arquivos apagados, deve-se especificar o
                       parâmetro "-v" ao comando, e "-i" para pedir
                       a confirmação ao apagar arquivos.
md        mkdir        Uma só diferença: No Linux permite que vários
                       diretórios sejam criados de uma só vez
                       (mkdir /tmp/a /tmp/b...)
copy      cp           Poucas diferenças. Para ser mostrados os arquivos
                       enquanto estão sendo copiados, deve-se usar a
                       opção "-v", e para que ele pergunte se deseja
                       substituir um arquivo já existente, deve-se usar
                       a opção "-i".
echo      echo         Sem diferenças
path      path         No Linux deve ser usado ":" para separar os
                       diretórios e usar o comando
                       "export PATH=caminho1:/caminho2:/caminho3:"
                       para definir a variável de ambiente PATH.
```

		O path atual pode ser visualizado através do comando "echo $PATH"
ren	mv	Poucas diferenças. No Linux não é possível renomear vários arquivos de uma só vez (como "ren *.txt *.bak"). É necessário usar um shell script para fazer isto.
type	cat	Sem diferenças
ver	uname -a	Poucas diferenças (o uname tem algumas opções a mais)
date	date	No Linux mostra/modifica a Data e Hora do sistema.
time	date	No Linux mostra/modifica a Data e Hora do sistema.
attrib	chmod	O chmod possui mais opções por tratar as permissões de acesso de leitura, gravação e execução para donos, grupos e outros usuários.
scandisk	fsck.ext2	O fsck é mais rápido e extensivo na checagem.
doskey	-----	A edição de teclas é feita automáticamente pelo bash.
edit	vi, ae, emacs	O edit é mais fácil de usar, mas usuário experientes apreciarão os recursos do vi ou o emacs (programado em lisp).
fdisk	fdisk, cfdisk	Os particionadores do Linux trabalham com praticamente todos os tipos de partições de diversos sistemas de arquivos diferentes.
format	mkfs.ext2	Poucas diferenças, precisa apenas que seja especificado o dispositivo a ser formatado como "/dev/fd0" ou "/dev/hda10" (o tipo de identificação usada no Linux), ao invés de "A:" ou "C:".
help	man, info	Sem diferenças
interlnk	plip	O plip do Linux permite que sejam montadas redes reais a partir de uma conexão via Cabo Paralelo ou Serial. A máquina pode fazer tudo o que poderia fazer conectada em uma rede (na realidade é uma rede e usa o TCP/IP como protocolo) inclusive navegar na Internet, enviar e-mails, irc, etc.
intersvr	plip	Mesmo que o acima.
keyb	loadkeys	Sem diferenças (somente que a posição das teclas do teclado pode ser editada. Desnecessário para a maioria dos usuários).
mem	cat /proc/meminfo top	Mostra detalhes sobre a quantidade de dados em buffers, cache e memória virtual (disco).
more	more, less	O more é equivalente a ambos os sistemas, mas o less permite que sejam usadas as setas para cima e para baixo, o que torna a leitura do texto muito mais agradável.
move	mv	Poucas diferenças. Para ser mostrados os arquivos enquanto estão sendo movidos, deve-se usar a

		opção "-v", e para que ele pergunte se deseja substituir um arquivo já existente deve-se usar a opção "-i".
scan	-----	Não existem virus no Linux devido as restrições do usuário durante execução de programas.
backup	tar	O tar permite o uso de compactação (através do parâmetro -z) e tem um melhor esquema de recuperação de arquivos corrompidos que já segue evoluindo há 30 anos em sistemas UNIX.
print	lpr	O lpr é mais rápido e permite até mesmo impressões de gráficos ou arquivos compactados diretamente caso seja usado o programa magicfilter. É o programa de Spool de impressoras usados no sistema Linux/Unix.
xcopy	cp -R	Pouca diferença, requer que seja usado a opção "-v" para mostrar os arquivos que estão sendo copiados e "-i" para pedir confirmação de substituição de arquivos.

Arquivos de configuração

Os arquivos config.sys e autoexec.bat são equivalentes aos arquivos do diretório /etc especialmente o /etc/inittab e arquivos dentro do diretório /etc/init.d .

Usando a sintaxe de comandos DOS no Linux

Você pode usar os comandos do pacote mtools para simular os comandos usados pelo DOS no Linux, a diferença básica é que eles terão a letra m no inicio do nome. Os seguintes comandos são suportados:

- mattrib - Ajusta modifica atributos de arquivos
- mcat - Mostra os dados da unidade de disquete em formato RAW
- mcd - Entra em diretórios
- mcopy - Copia arquivos/diretórios
- mdel - Exclui arquivos
- mdeltree - Exclui arquivos, diretórios e sub-diretórios
- mdir - Lista arquivos e diretórios
- mdu - Mostra o espaço ocupado pelo diretório do DOS
- mformat - Formatador de discos
- minfo - Mostra detalhes sobre a unidade de disquetes
- mlabel - Cria um volume para unidades DOS
- mmd - Cria diretórios
- mmount - Monta discos DOS
- mmove - Move ou renomeia arquivos/subdiretórios
- mpartition - Particiona um disco para ser usado no DOS
- mrd - Remove um diretório
- mren - Renomeia arquivos
- mtype - Visualiza o conteúdo de arquivos (equivalente ao cat)
- mtoolstest - Exibe a configuração atual do mtools

- mshowfat - Mostra a FAT da unidade
- mbadblocks - Procura por setores defeituosos na unidade
- mzip - Altera modo de proteção e ejeta discos em unidades Jaz/ZIP
- mkmanifest - Cria um shell script para restaurar nomes extensos usados no UNIX
- mcheck - Verifica arquivos na unidade

Programas equivalentes entre Windows/DOS e o Linux

Esta seção contém programas equivalentes para quem está vindo do DOS e Windows e não sabe o que usar no Linux. Esta seção também tem por objetivo permitir ao usuário que ainda não usa Linux decidir se a passagem vale a pena vendo se o sistema tem os programas que precisa. Note que esta listagem mostra os programas equivalentes entre o DOS/Windows e o Linux cabendo a você a decisão final de migrar ou não. Lembrando que é possível usar o Windows, OS/2, DOS, OS/2 e Linux no mesmo disco rígido sem qualquer tipo de conflito.

DOS/Windows	Linux	Diferenças
MS Word	Star Office, Corel Word Perfect	O Star Office possui todos os recursos do Word além de ter a interface gráfica igual, menus e teclas de atalho idênticas ao Word, o que facilita a migração. Também trabalha com arquivos no formato Word97/2000 e não é vulnerável a virus de macro. É distribuído gratuitamente e não requer pagamento de licença podendo ser instalado em quantos computadores você quiser (tanto domésticos como de empresas).
MS Excel	Star Office	Mesmos pontos do acima e também abre arquivos Excel97/2000.
MS PowerPoint	Star Office	Mesmos pontos do acima.
MS Access	SQL, Oracle, etc	Existem diversas ferramentas de conceito para bancos de dados corporativos no Linux. Todos produtos compatíveis com outras plataformas.
MS Outlook	Pine, Mutt, etc	Centenas de programas de E-Mail tanto em modo texto como em modo gráfico. Instale, avalie e escolha.
MS Internet Explorer	Netscape, Arena, Mozilla, lynx.	Os três primeiros para modo gráfico e o lynx opera em modo texto.
ICQ	LICQ	Muito prático e fácil de operar. Possibilita a mudança completa da aparência do programa

		através de Skins. A organização dos menus deste programa é outro ponto de destaque.
Photo Shop	The Gimp	Fácil de usar, possui muitos scripts que permitem a criação rápida e fácil de qualquer tipo de efeito profissional pelo usuário mais leigo. Acompanha centenas de efeitos especiais e um belo manual em html com muitas fotos (uns 20MB no total)que mostra o que é possível se fazer com ele.
Corel Photo Paint	Corel Photo Paint	Corel Photo-Paint para Linux.
winamp	xmms	Possui todos os recursos do programa para Windows além de filtros que permite acrescentar efeitos digitais da música (em tempo real), eco, etc.
media player	xanim, xplaymidi xwave,	Programas para execução de arquivos de música e videos multimídia. Existem outras alternativas, a escolha depende de seu gosto e da sofisticação do programa.
Agente de Sistema	cron	Pouca diferença. O cron da mais liberdade na programação de tarefas a serem executadas pelo Linux.
Mixer	aumix, cam	Sem diferenças.
Bate-Papo	talk, ytalk	O talk e o ytalk permite a conversa de dois usuários não só através de uma rede local, mas de qualquer parte do planeta, pois usa o protocolo tcp/ip para comunicação. Muito útil e fácil de usar.
MIRC	Bitchx, xchat	Clientes IRC para Linux
Frontpage Server	apache	Sem comentários, o apache é o servidor WEB mais usado no mundo (algo em torno de 75% das empresas), muito rápido e flexível de se configurar.
Exchange, NT Mail	sendmail, smail qmail	Só o sendmail tem uma base instalada de mais de 70% no

		mundo. o Smail é o mais rápido e o qmail é o mais seguro. Todos (especialmente o sendmail) tem como característica a flexibilidade de configuração.
Wingate, MS Proxy	squid, apache, ip masquerade, nat, diald, smail,	A migração de um servidor proxy para Linux requer o uso de vários programas separados para que se tenha um resultado profissional. Isto pode parecer incomodo no começo, mas você logo perceberá que a divisão de serviços entre programas é mais produtivo. Quando desejar substituir um deles, o funcionamento dos outros não serão afetados. Não vou entrar em detalhes sobre os programas citados ao lado, mas o squid é um servidor proxy Web (HTTP e HTTPS) completo e também apresenta um excelente serviço FTP. Possui outros módulos como dns, ping, restrições de acesso, limites de tamanho de arquivos, cache, etc.
MS Frontpage	Netscape Composer e muitas outras ferramentas para geração de conteúdo WEB (como zope, php3, php4, wdm, htdig)	Sem comentários... todas são ferramentas para a geração de grandes Web Sites. O wdm, por exemplo, é usado na geração do site da distribuição Debian (http://www.debian.org) em 27 idiomas diferentes.
MS Winsock	Sem equivalente	O Linux tem suporte nativo a tcp/ip desde o começo de sua existência e não precisa de nenhuma camada de comunicação entre ele e a Internet. A performance é aproximadamente 10% maior em conexões Internet via fax-modem.
ViruScan, TBAV, F-PROT, CPAV.	-----	Não existem vírus no Linux devido as restrições ao usuário durante a execução de programas.